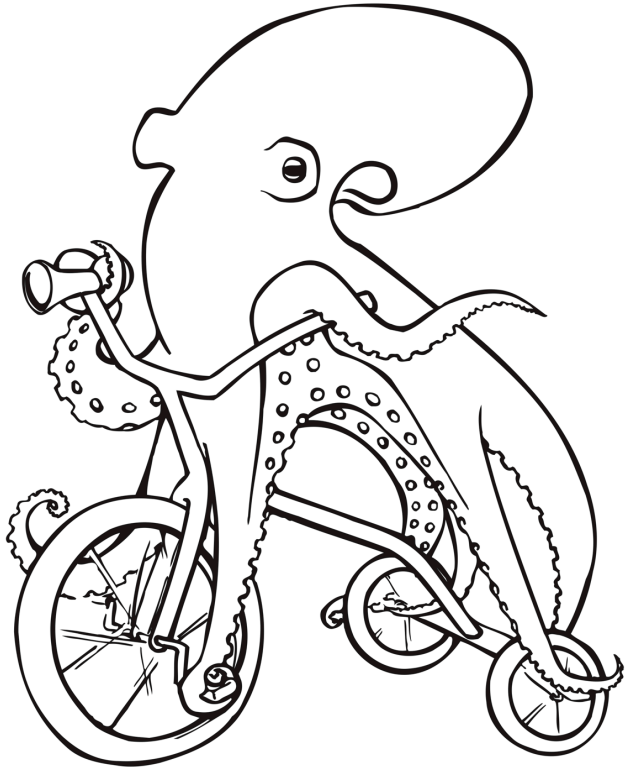
Threat Modeling Using Trike

Methodology Overview

Brenda Larcom
February 27, 2012

http://www.octotrike.org/



STACH&LIU

Agenda

What are we going to talk about?

- Differences
- Theory
- Anatomy
- Creation
- Use
- Tool Support

Differences

What's so cool about Trike?

- Generate threats [semi]-automatically, no brainstorming
- Security-inexperienced developers reliably find issues
- Security geeks can pick up where developers left off
- It's clear what to analyze
- It's clear when to stop
- Attack chaining, not attack trees
- Tools provide immediate feedback as you design
- Start earlier, with requirements
- Include sequences of events, not just static architecture
- Include intended system behavior

Differences

What's the catch?

- Tools & methodology hard-code theory
- Heavily reliant on automation
- All available tools are bleeding edge
- Requires more data about the system
- Different, more restrictive definitions
- It's clear when you've stopped too soon

Theory

What are Trike's basic assumptions?

- Models are for answering questions
- Threat models can only answer technical questions
- Developers know about the system, security geeks know about security
  - The reverse may not be true
- Secure enough = meets security objectives
- Attacker goals are irrelevant
- Threats = f(system)
- Attackers will use both intended & unintended system behavior

Anatomy

What goes into a threat model?

Systems Analysis

- Functional requirements
  - Actors
  - Assets
  - Intended actions
- Deployment environment
- Security objectives
  - In-scope requirements threats
  - In-scope attackers
- System architecture
  - Static view
  - Dynamic view
  - Security attributes & technology

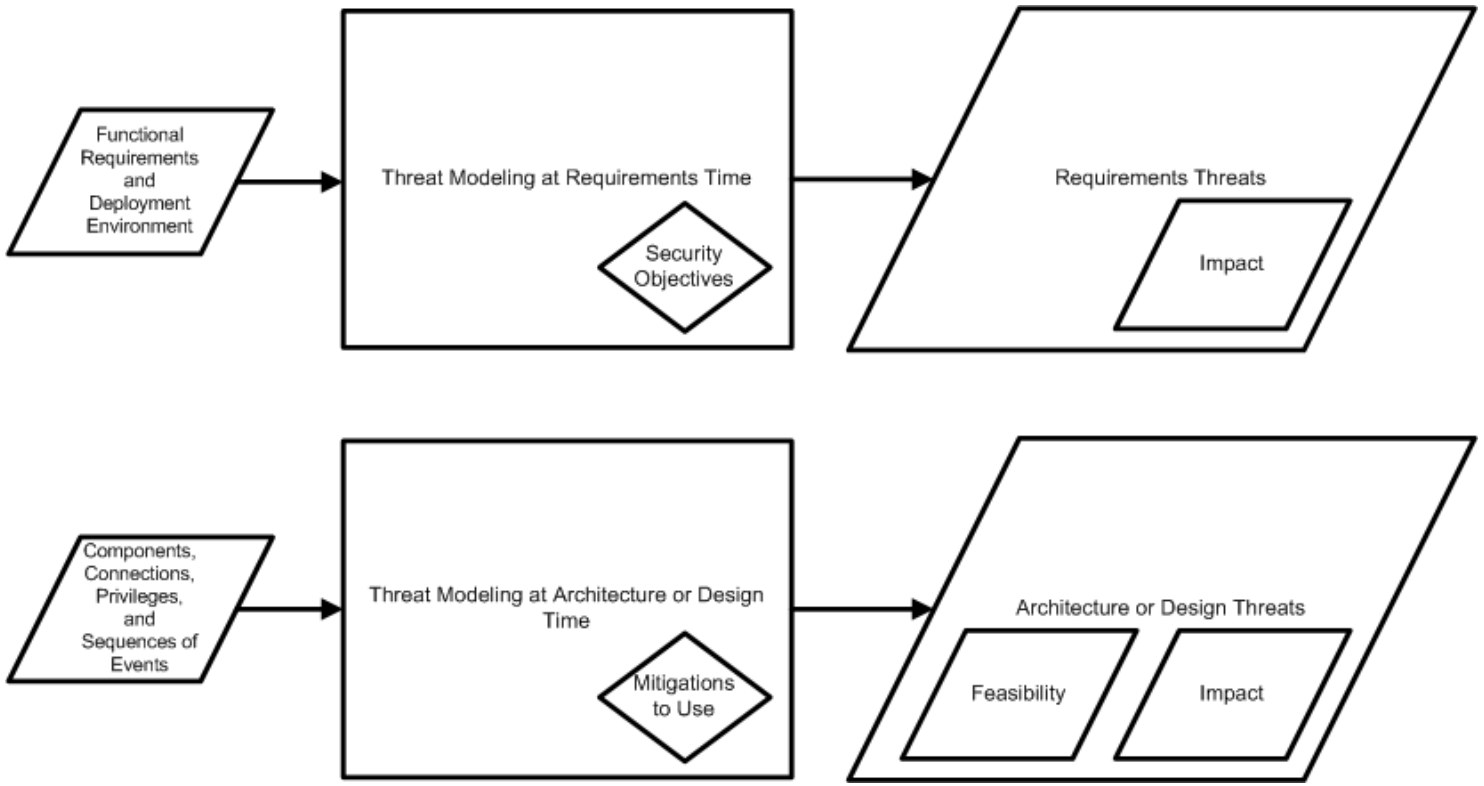
- Requirements threats
- Architecture threats
  - Feasibility
- Relationships between threats
  - Impact
- Mitigations
  - Effectiveness

Security Analysis

STACH&LIU

Creation

What do I do?



Creation

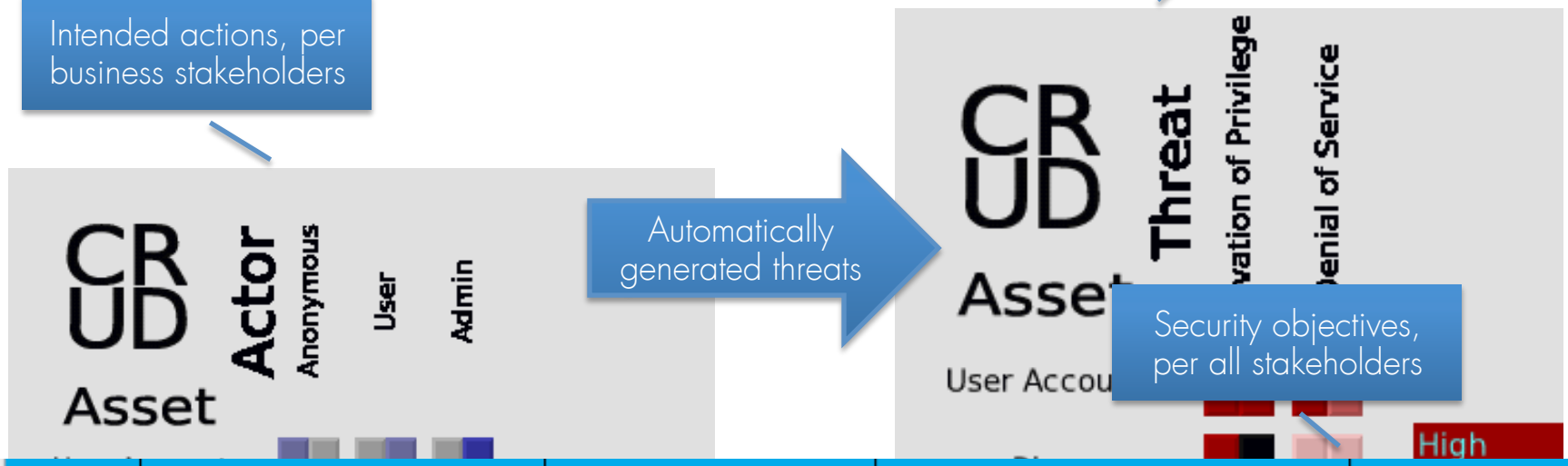
What happens at requirements time?

Intended actions, per business stakeholders

Automatically generated threats

Threat impact, per business stakeholders

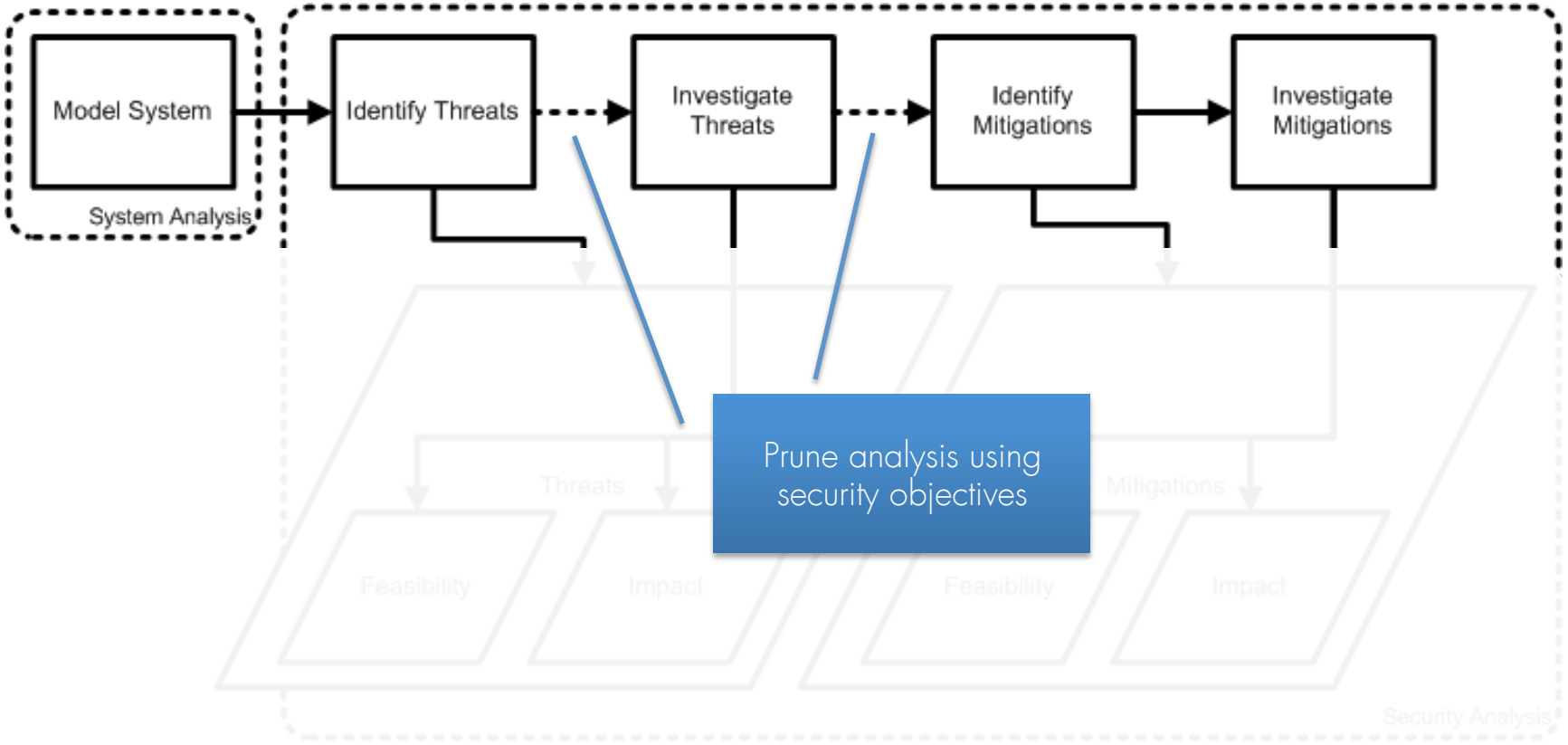
Security objectives, per all stakeholders



#	Initial Configuration	Attackers	Prohibited Threats	Intended Response
SO1	Blog exists	Anonymous	Create Blog Entry on someone else's Blog	Thwart
	Blog is not configured to allow posting by the attacking user	User	Delete Blog Entry	Detect and Log

Creation

What do/did I do?



Creation

What happens at architecture time?

- Privilege analysis
  - Finds inconsistencies and issues in privileges a component or connection has, provides, revokes, uses, and requires
  - Likely automatable; theory still in development
- HAZOP analysis
  - Structured analysis technique from safety engineering
  - Identifies harmful variations in sequences of events
  - Semi-automatable
- Attack chaining
  - Collections of privileges are nodes, intended and unintended behaviors are edges
  - Definitely automatable
  - Need to investigate scaling/computational complexity issues
  - Prototype in development

You can do this now

Creation

What does a sequence of events look like?

Use Case #	Step #	Path		Terminal	Actor	Action	Object	Condition
		Choice	Choice					
UC3	1				User	submits	Blog Entry	to Web Server
	2				AV Filter	scans	Blog Entry	
	3	Virus			Web Server	sends	Status Response	to User
								reflecting failure due to virus
		No Virus			Blog Module	compares	User Account	to Blog Permissions
	4	No Virus	Authorized		Blog Module	creates	Blog Entry	in Database
	5	No Virus	Authorized		Blog Module	creates	Log Entry	in Database
	6	No Virus	Authorized		Blog Module	sends	Status Response	to User
								reflecting success

Creation

How do I vary a step?

Use Case #	Step #	Condition	Varied Element	Guide Word	Guide Word Meaning	Variation
UC3	1	to Server	Actor	NO	Actor is not in the correct role, or does not have the capability.	A User who is not logged in submits a Blog Entry to the Web Server.
			Actor	AS WELL AS	Actor is in the correct role, but is also in another (typically more privileged) role or otherwise has additional capabilities.	A User who is also an Admin submits a Blog Entry to the Web Server.
			Actor	PART OF	Actor has some, but not all of the needed capabilities.	The attacker submits a Blog Entry to the Web Server from a public terminal a User just logged out of.

Creation

How do I analyze a variation?

Use Case #	Step #	Variation	Security Objectives Variation Would Help Attacker Achieve: SO1						Rationale for Variation's Helpfulness to Attacker	Attacker Influenced	Rationale for Attacker Influenced	Issue Criticality
UC3	1	A User who is not logged in submits a Blog Entry to the Web Server.							There is no step that checks whether the User is actually logged in; the Web Server will accept any Blog Entry sent to this interface.		There's nothing stopping anyone on the Internet from submitting a Blog Entry.	High
		A User who is also an Admin submits a Blog Entry to the Web Server.							Our security objectives trust all Admins.			
		The attacker submits a Blog Entry to the Web Server from a public terminal a User just logged out of.							On logout, the server invalidates the User's session and instructs the client to delete all cookies; a terminal the User logged out of is no more useful than a terminal the User has never used.			

Use

How do I use a threat model to make decisions?

- Identify a project decision that should be affected by security
  - E.g. Whether application is ready to launch
- Identify information that should inform that decision
  - E.g. Does the expense reports application meet its security objectives?
- Extract that information from the model
  - E.g. Examine threats that are still feasible for unbroken chains from attacker starting privileges to prohibited threats

Use

How do I use a threat model at design time?

- Security objectives should be met
- Defenses should be protecting against threats
- Apply design patterns appropriately to respond to threats (e.g. input trust boundary, centralized input validation library)
- Best design has either fewer or easier threats to defend against

Use

How do I use a threat model to drive security tests?

- Confirm protections are in place
- Confirm responsibilities are met
- Try to perform all the relevant threats identified in the threat model
    - Start with those that are more beneficial to the attacker

Tool Support

What can I have Right Now?

- Trike 1, in Squeak
  - Auto-generates threats based on intended actions & lets you prioritize them
  - Auto-generates attack tree stubs (deprecated)
  - No file import or export
- Trike 1.5, as a spreadsheet
  - Auto-generates threats based on intended actions and deployment environment & lets you prioritize them
  - Security objectives
  - Data collection, but no analysis (yet) for component & connection privileges
  - Data collection & basic support for HAZOP analysis
  - Updated regularly

http://www.octotrike.org/

Tool Support

Where is this headed?

- Trike 2, in Squeak
  - Have some code, in re-design now
  - Will implement everything discussed here
  - Sketch-based interface that highlights problems and missing information as you draw
  - REST interface in case you hate our futuristic UI enough to write a different one
  - Yes, it will do files, I promise
  - No firm ETA yet, but 2013 is more likely than 2012
  - Security objectives portion will likely come out first

Thanks

- Eleanor Saitta
- Erik Simmons
- Khyati Shrivastava

- Mozilla!

For more information, see http://www.octotrike.org/.